



# Liquibase

...

<http://www.liquibase.org/>

# Our Sources, our Source Code

We use a **V**ersion **C**ontrol **S**ystem to manage changes in our code

## Benefits

1. Collaboration, working together on the code (‘pull requests’ in github)
2. Storing versions and Creating releases
3. Understanding what is changing ( `git commit -m “fixed issue-4”` )
4. Restoring previous version(s) ( *makes us braver* )
5. Backup of your source code

A VCS gives us more than one benefit

## Other benefits

1. **Local**: Developers are cloning/pulling the latest changes to their local machines
2. **Test**: We are cloning/pulling the latest changes to the Test environment (Alternative , we are pulling release XX)
3. **Prod**: We are cloning/pulling the latest changes to the Production environment (Alternative , we are pulling release XX)

## Our Sources, our Source Code

It is more difficult to manage changes in a database

We would like to manage database changes in the same way as

we do in traditional source code

Our Sources, our Source Code

Liquibase is a tool for  
‘database change management’

In Liquibase,

We store our db-changes in textfile(s) (DDL and DML),  
we then apply them ‘manually’ to the database ...  
with the Liquibase tool

# database change management

## In Liquibase

We write one or many 'changelog'-files,  
changelog-files contain changesets

An example project :

<https://github.com/Inkimar/liquibase-tutorial>

<https://github.com/Inkimar/liquibase-tutorial>

16 lines (15 sloc) | 796 Bytes

```
1 <databaseChangeLog
2     xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
5     xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangel
6     http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.li
7 <changeSet id="1" author="test-1">
8     <createTable tableName="debug_info">
9         <column name="message" type="varchar(255)" />
10        <column name="date_ran" type="datetime">
11            <constraints nullable="false" />
12        </column>
13    </createTable>
14 </changeSet>
15 </databaseChangeLog>
```

### Some Rules

1. Every changeSet has a unique identifier ( id+author )
2. Every changeSet contains a minimal change.
3. You **never** change a changeSet, if you ran a 'faulty' changeset then you add a new 'fix'-changeset to make the wrong right ....  
( <http://www.liquibase.org/bestpractices.html> )

<https://github.com/Inkimar/liquibase-tutorial>

## The 'liquibase-tutorial'- project

- tested on GNU/Linux with MySQL

Before running liquibase , you **must** have created the 'liquibase-tutorial'-db + a .env-file with credentials ( see .env\_template)

The Makefile contains recipes for

1. Downloading the liquibase zip-file
2. Downloading the mysql-jdbc-jar
3. Running liquibase on the 'changelog-example.xml'-file
4. Database manipulation ('create','drop','mysqldump')



# database change management

The core of a liquibase project is the database changelog-file,

it is possible to break up the changelogs into multiple files. See the following project

<https://github.com/DINA-Web/collections-data-model-docker>

The db.changelog-master.xml contains

1. tables.sql
2. optional.sql

# database change management

formats for writing changesets : XML, YAML, JSON  
or SQL

Paul morris uses the SQL-syntax in the 2 files below

<https://github.com/DINA-Web/collections-data-model-docker>

The db.changelog-master.xml contains  
references to

1. tables.sql
2. optional.sql

Liquibase and VCS gives us benefits

- 1.Local: All developers have the same database-schema
- 2.Test: The same database-schema
- 3.Prod: The same database-schema

Liquibase and A Version Control System  
helps us to be Agile

# Another benefit ....

You can become Database Agnostic

supported databases :

<http://www.liquibase.org/databases.html>

Database	Type Name	Notes
MySQL	mysql	No Issues
PostgreSQL	postgresql	8.2+ is required to
Oracle	oracle	11g driver is required or AL16UTF16
Sql Server	mssql	No Issues
Sybase_Enterprise	sybase	ASE 12.0+ required JTDS. Sybase does not support foreign key functionality.
Sybase_Anywhere	asany	Since 1.9
DB2	db2	No Issues. Will a
<a href="#">Apache_Derby</a>	derby	No Issues
HSQL	hsqldb	No Issues
H2	h2	No Issues
<a href="#">Informix</a>	informix	No Issues
Firebird	firebird	No Issues
<a href="#">SQLite</a>	sqlite	No Issues

# Liquibase/about

<http://www.liquibase.org/about.html>

## Source Code

All source code managed as part of the Liquibase project is available at <https://github.com/liquibase>

The main Liquibase code is at <https://github.com/liquibase/liquibase>

## License

Liquibase is licensed under the Apache 2.0 License

## Friends of Liquibase

The following companies have provided software and/or services to help support Liquibase:

- **Datical** (Sponsors Liquibase Development <http://datical.com>)
- **Atlassian** (Provides <http://liquibase.jira.com>)
- **Zoho** (Provides <http://forum.liquibase.org>)
- **YourKit** (Provides [Java Profiler](#))

## Open Source Projects Using Liquibase

- [Portofino by ManyDesigns](#)
- [JHipster](#)
- Know one to add? [Let us know](#)

# Other tools , - I have not looked into

1.Flyway <https://flywaydb.org/>

2.dbdeploy : <http://dbdeploy.com/>

3.dbmaintain : <http://www.dbmaintain.org/>

And other tools I do not know that exists ?

## Tools to look into ?

4.dbunit : <http://dbunit.sourceforge.net/>

5.sqlunit : <http://sqlunit.sourceforge.net/>

<https://github.com/Inkimar/liquibase-tutorial>

The zip-file comes from ->

<http://www.liquibase.org/download/index.html>

Documentation : <http://www.liquibase.org/documentation/index.html>

interesting notes on 'context' -

- I have **not** checked out the 'context'
- I am not an fully educated on Liquibase